Webology, Volume 18, Number 2, 2021
ISSN: 1735-188X
DOI: 10.29121/WEB/V18I2/52

Designing and Validation of Block Random Access Memory based on Hardware Descriptive Language

Vineet Singh Deopa[1], Kamlesh Kukreti[1], Alankrita Joshi[1], Dr Vrince Vimal[2]

[1]Graphic Era Deemed To Be University, Dehradun, Uttarakhand, India
[2]Professor, Department of Computer Science and Engineering, Graphic Era Hill University, Dehradun

ABSTRACT

Today every device has an on chip memory associated with it and therefore it is essential for everyone and to efficiently utilize our memory actually plays an important role. BRAM is an important memory block inside FPGA which can be configured as a multiported memory as per our need and requirement. On chip memories have become an important part of every electronic device, therefore there is a need to efficiently design the on chip memories. In this paper I have represented the RTL schematic of a BRAM and then the synchronous read and write operation can be seen in the simulation results.

Keywords: BRAM (Block Random Access Memory), FPGA (Field Programmable Gate Array), CLB (Complex Logic Block), WR En (Write Enable), RD En (Read Enable), CLK (Clock).

INTRODUCTION

Virtual BRAM stands for Block Random Access Memory. BRAMs are used for storing large amount of data inside FPGA. The bigger and more expensive the FPGA the more number of BRAM it will have on it. As per our need and requirement we can configure BRAM as a single or as a dual port RAM. When BRAM is configured as a single port RAM then we can either read or write data into it at a time, so when BRAM is configured as a single port data can be read or written only once no synchronous read and write operation takes place. In a dual port mode, we can perform synchronous read and write operations simultaneously. With FPGA we can implement various types of storage memory. An FPGA consists of large array of CLBs. Each CLB represents almost any four section of combinational or sequential logic. The configuration of a BRAM as a true dual port means it has two independent read and write ports and these ports can be read or written simultaneously, independent of each other. All control logic is implemented within the BRAM, so no CLB is required to implement dual port configuration [4].

The internal architecture of BRAM consists of CLBs which consists of flip-flops, Look Up Tables (LUTs) and multiplexers [5].

Complex Logic Block (CLB): A CLB is the basic repeating blocks inside an FPGA, when CLBs are grouped together by means of routing resources the components inside CLB execute complex logic functions, memory functions and they are used to run the code inside an FPGA.

The components of CLBs contain mainly three core elements
Flip Flops: A sequential circuit which consists of two stable states that represents a single bit of information. A flip flop is the smallest storage element inside an FPGA.

Look Up Tables (LUTs): A look up table is used to store predefined list of outputs for each combination of inputs.

Multiplexer: A combinational circuit in which any one of the input is reflected at the output by means of select lines. A multiplexer is often called many to one circuit.

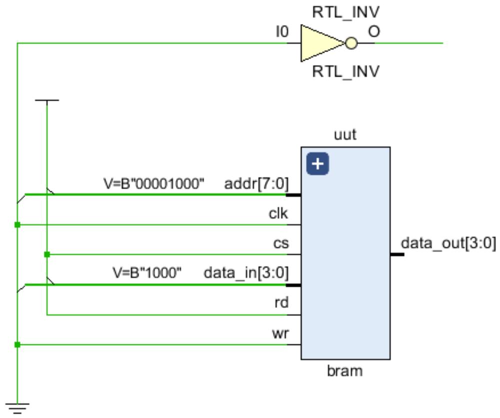


Fig 1: Block diagram of a BRAM

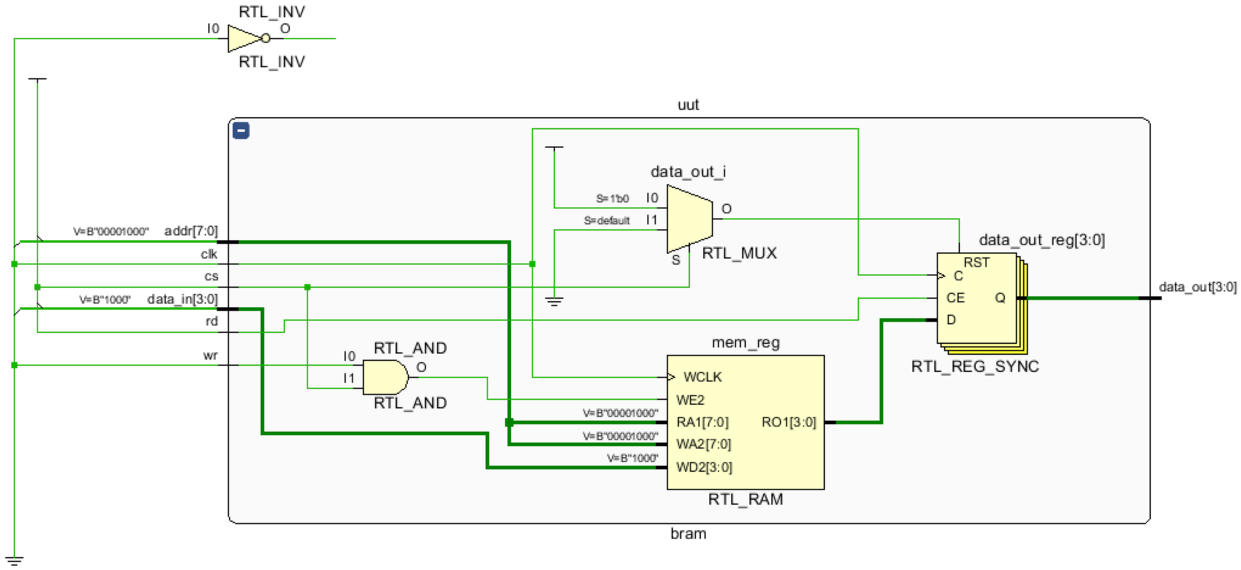


Fig 2: RTL schematic of a BRAM

DESIGN OF A FOUR BIT BRAM

BRAM is required to store large amount data which is helpful for instruction execution purpose. The four bit BRAM consists of 4-bit wide input and output data with 8bit of address locations. To perform only write operation inside a BRAM CLK, CS and WR signal needs to be high with RD signal to be low similarly when only read operation is performed CLK, CS and RD signal needs to be high with WR signal to be low [1]. When synchronous read and write operations needs to be performed then all the signals needs to be high and these operations can be seen in the simulation results. BRAM has two completely independent ports [3].

There are two ports that support data read and write operations. When BRAM is operated in a dual port mode then both the ports behave as an independent ports supporting simultaneous read and write operations at different address locations [5].

When no operation takes then no data will be written or read into the memory and initially the data will be 0000 at address location 00000000 and as soon as either write signal goes high read port is disabled and write operation in a BRAM starts taking place [6].

Table 1: Write operation being performed in a BRAM

CLK	CS	WR	RD	Addr[7:0]	Data_in[3:0]	Data_out[3:0]
0	0	0	0	00000000	0000	0000
1	1	1	0	00000001	0001	0000
0	1	1	0	00000010	0010	0000
1	1	1	0	00000011	0011	0000
0	1	1	0	00000100	0100	0000
1	1	1	0	00000101	0101	0000
0	1	1	0	00000110	0110	0000
1	1	1	0	00000111	0111	0000
0	1	1	0	00000100	1000	0000

Table 2: Write operation being performed in a BRAM

CLK	CS	WR	RD	Addr[7:0]	Data_in[3:0]	Data_out[3:0]
0	0	0	0	00000000	1000	0000
1	1	0	1	00000001	1000	0001
0	1	0	1	00000010	1000	0010
1	1	0	1	00000011	1000	0011
0	1	0	1	00000100	1000	0100
1	1	0	1	00000101	1000	0101
0	1	0	1	00000110	1000	0110
1	1	0	1	00000111	1000	0111
0	1	0	1	00000100	1000	1000

DESIGN OF EXPERIMENT/ MATERIAL METHODS (Times New Roman, 14)

In order to measure the volatileness of the series of time, the models of ARCH that were first brought under proposition by Engel on 1982 became the most widely referred models to evaluation of stock market values. (Paragraph, Times New Roman, 12)

SIMULATION RESULTS AND DISCUSSION

Xilinx Vivado was used for the functional verification of the design with an inbuilt simulator, with the help of simulator we can check the simulated waveform output by performing behavioral simulation in Xilinx Vivado. The inbuilt simulator of Xilinx Vivado helps us to perform the behavioral simulation of the DUT. The stimulus or test bench so created helps us to verify the functionality of the design that with respect to the input stimulus what will be the simulated waveform at the output. As per our need and requirement whether we want to perform read, write or synchronous read and write operations [7]. When the write signal goes high at that time we can see write operation at the simulated waveform output at different address locations as shown in Tabe 1(a). We can clearly see in the simulated waveform as shown in Fig 2. When write operation is in process WR signal goes high and write port is enabled and read port is disabled so in this case data is written into different address locations. Data will be written until the write signal is high. At address location 00000000 initially data is 0000 but as the data starts being written into the memory then as we can see the write operation is being performed at address location 00000001 data written is 0001 and at address location 00000010 data written is 0010 and at location 00000011 data written is 0100 and at location 00000100 data written is 1000 similarly the process goes on for write operation.

Now when the read operation takes place in this case write signal goes low and write port is disabled, now read operation starts taking place so in this case read signal goes high and read port is enabled and data is read from the memory at different address locations which can be seen in the simulated waveform output at Fig 3 and also the read operation and fetching of data can also be seen in Table 1(b) and the synchronous read and write operation starts taking place. So from the simulation results we observed that how synchronous read and write operation takes place inside a BRAM [8].

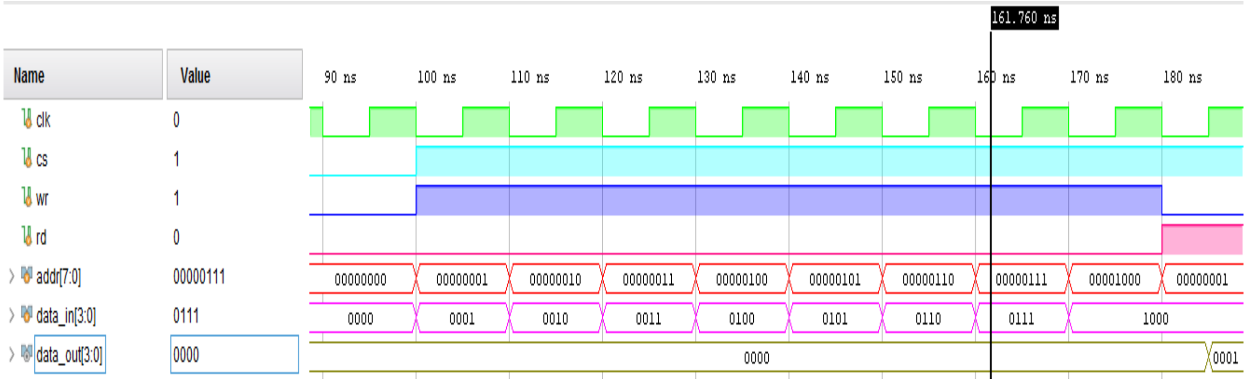
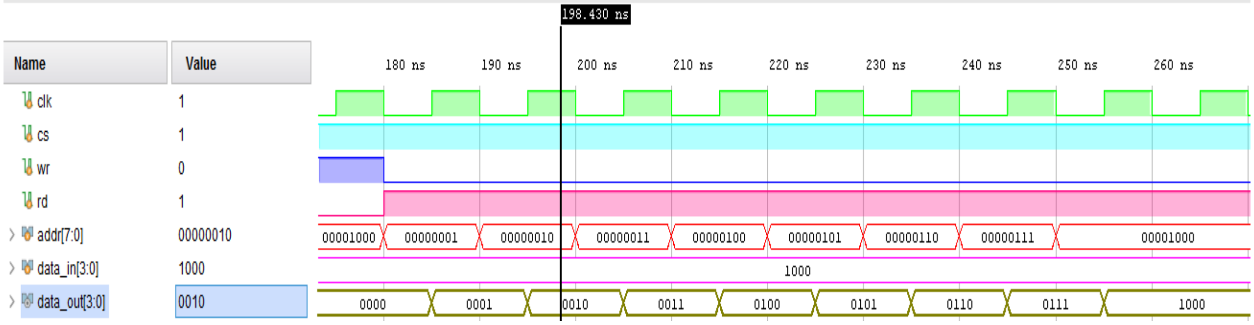


Fig 2: Write operation being performed in a BRAM



Name	Value
clk	1
cs	1
wr	0
rd	1
addr[7:0]	00000010
data_in[3:0]	1000
data_out[3:0]	0010

Fig 3: Read operation and fetching of data being performed in a BRAM

CONCLUSION

The design and functionality have been verified by performing the behavioral simulation of the DUT using structural level of modeling. The BRAM design which I carried out is the base of the complex designs which are widely being used in the industry and the knowledge of base design actually plays an important role in designing the new age complex architectural models. As a part of the future enhancement to the DUT using Verilog HDL we can implement the design methodology to design different on chip memories such as DRAM and SRAM. To enhance the functionality of the design to the next step we can implement this design using System Verilog methodology,

REFERENCES

1. Kang, S. M., & Leblibici, Y. (2003). CMOS (Third edition). Tata McGraw Hill.Kiran, P. V., & Saxena, N. (2015, February). Design and analysis of different types SRAM cell topologies. In Electronics and Communication Systems (ICECS), 2015 2nd International Conference on (pp. 1060-1065). IEEE.
2. Mondal, P.; Biswal, P.K.; Banerjee, S. FPGA based accelerated 3D affine transform for real-time image processing applications. Comput. Electr. Eng. 2016, 49, 69–83. [CrossRef]
3. Wang, W.; Yan, J.; Xu, N.; Wang, Y.; Hsu, F.H. Real-Time High-Quality Stereo Vision System in FPGA. IEEE Trans. Circuits Syst. Video Technol. 2015, 25, 1696–1708. [CrossRef]
4. Jin, S.; Cho, J.; Pham, X.D.; Lee, K.M.; Park, S.K.; Kim, M.; Jeon, J.W. FPGA Design and Implementation of a Real-Time Stereo Vision System. IEEE Trans. Circuits Syst. Video Technol. 2010, 20, 15–26.
5. Perri, S.; Frustaci, F.; Spagnolo, F.; Corsonello, P. Design of Real-Time FPGA-based Embedded System for Stereo Vision. In Proceedings of the 2018 IEEE International Symposium on Circuits and Systems (ISCAS), Florence, Italy, 27–30 May 2018; pp. 1–5.
6. Stevanovic, U.; Caselle, M.; Cecilia, A.; Chilingaryan, S.; Farago, T.; Gasilov, S.; Herth, A.; Kopmann, A.; Vogelgesang, M.; Balzer, M.; Baumbach, T.; Weber, M. A Control System and Streaming DAQ Platform with Image-Based Trigger for X-ray Imaging. IEEE Trans. Nucl. Sci. 2015, 62, 911–918.
7. Gallo, L.; Cilardo, A.; Thomas, D.; Bayliss, S.; Constantinides, G.A. Area implications of memory partitioning for high-level synthesis on FPGAs. In Proceedings of the 2014 24th International Conference on Field Programmable Logic and Applications (FPL), Munich, Germany, 2–4 September 2014; pp. 1–4.

8. Cilardo, A.; Gallo, L. Interplay of Loop Unrolling and Multidimensional Memory Partitioning in HLS. In Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition (DATE '15), Grenoble, France, 9–13 March 2015; EDA Consortium: San Jose, CA, USA; pp. 163–168.
9. Wang, Y.; Li, P.; Zhang, P.; Zhang, C.; Cong, J. Memory Partitioning for Multidimensional Arrays in High-level Synthesis. In Proceedings of the 50th Annual Design Automation Conference, Austin, TX, USA, 29 May–7 June 2013; pp. 12:1–12:8.
10. Kukreti Kamlesh, Kumar Prashant, Barthwal Shivangi, Juyal Amit, Joshi Alankrita, "Performance Analysis of Full Adder based on Domino Logic Technique", 2021 6th International Conference on Inventive Computation Technologies (ICICT). IEEE Xplore, 312-316.